

Microprocessor **PWM & ADC**



Yu Hong Sun

Goal

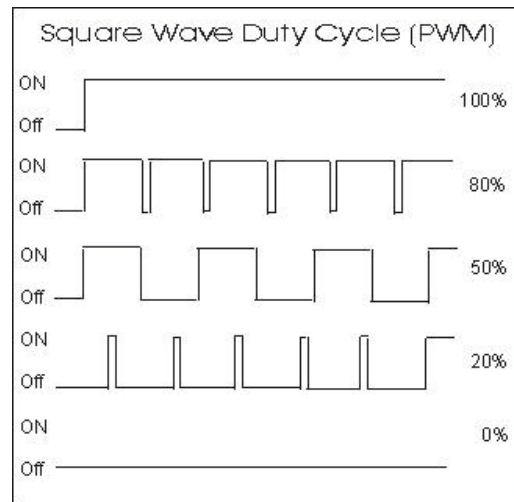
◆ 목표

- 타이머/카운트를 활용하여 PWM을 만들어 보자.
- ADC를 이해하고, 활용하여 보자.

Pulse Width Modulation

◆ PWM이란?

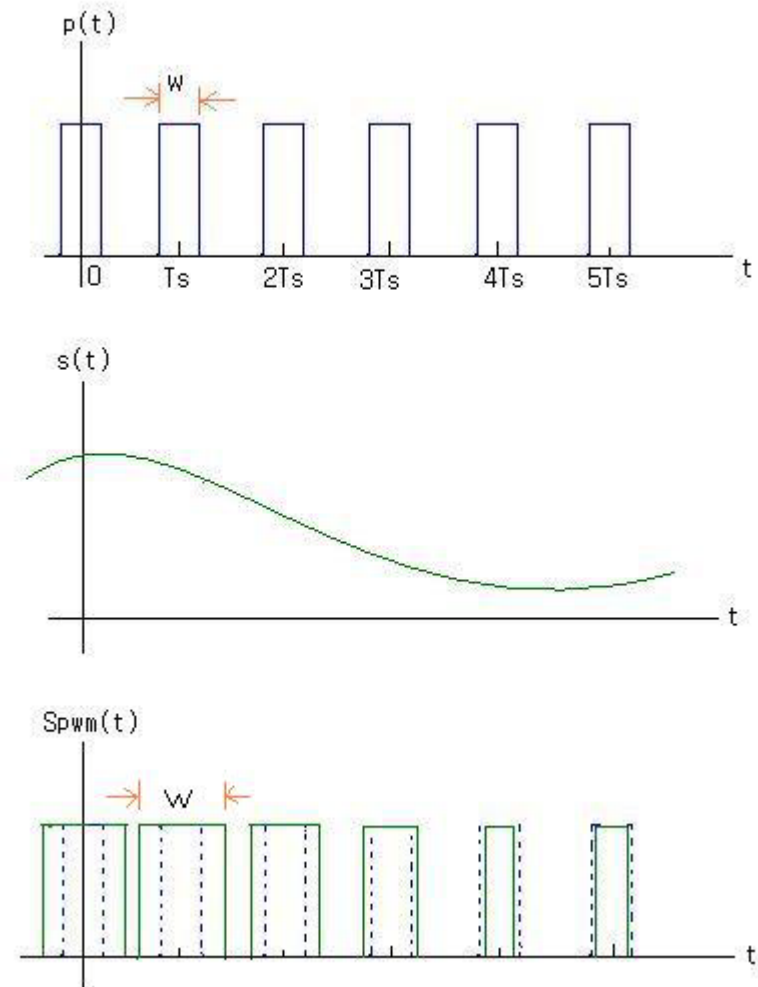
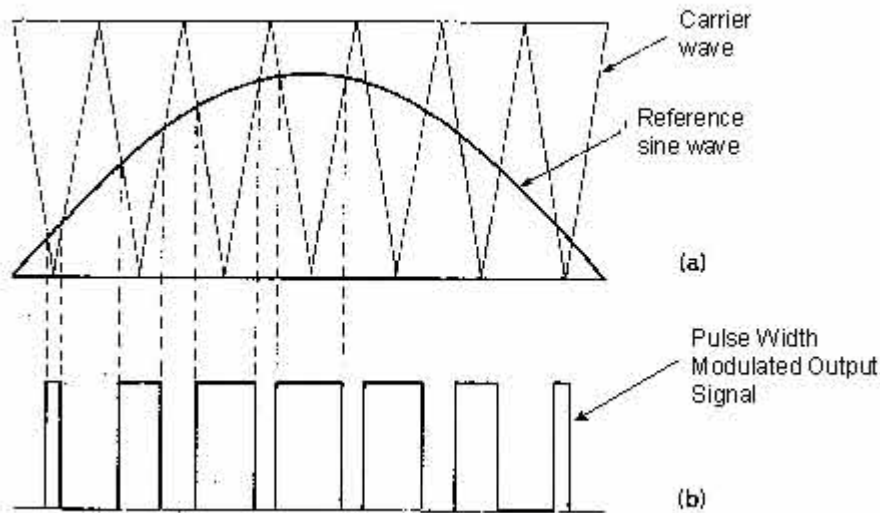
- Pulse Width Modulation
- 표본화 펄스의 진폭을 일정하고 그 펄스 폭을 전송하고자 하는 신호에 따라 변화 시키는 변조 방식



“디지털 출력으로 아날로그 회로를 제어하는 강력한 기법”

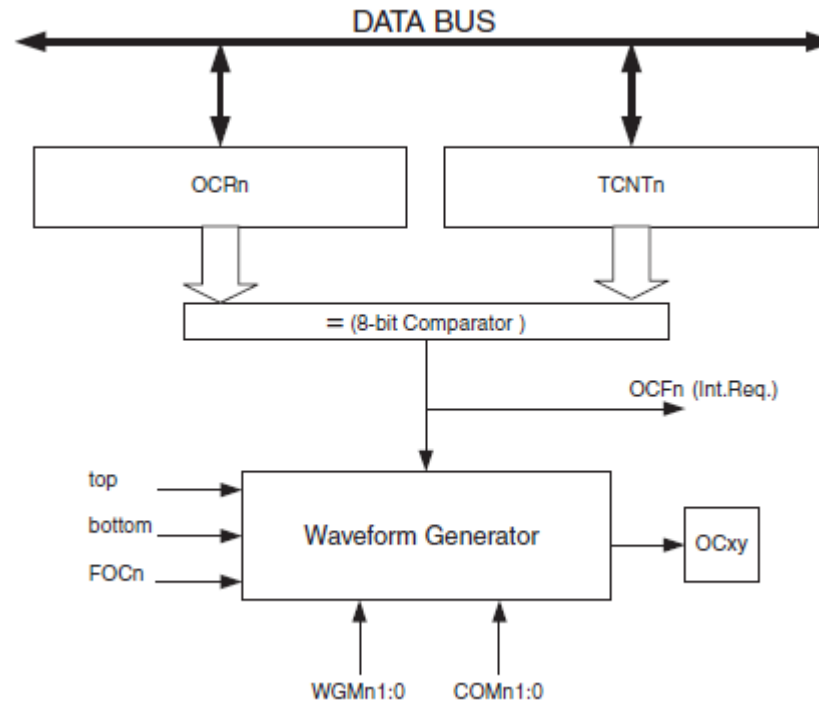
Pulse Width Modulation

◆ PWM 만드는 방법



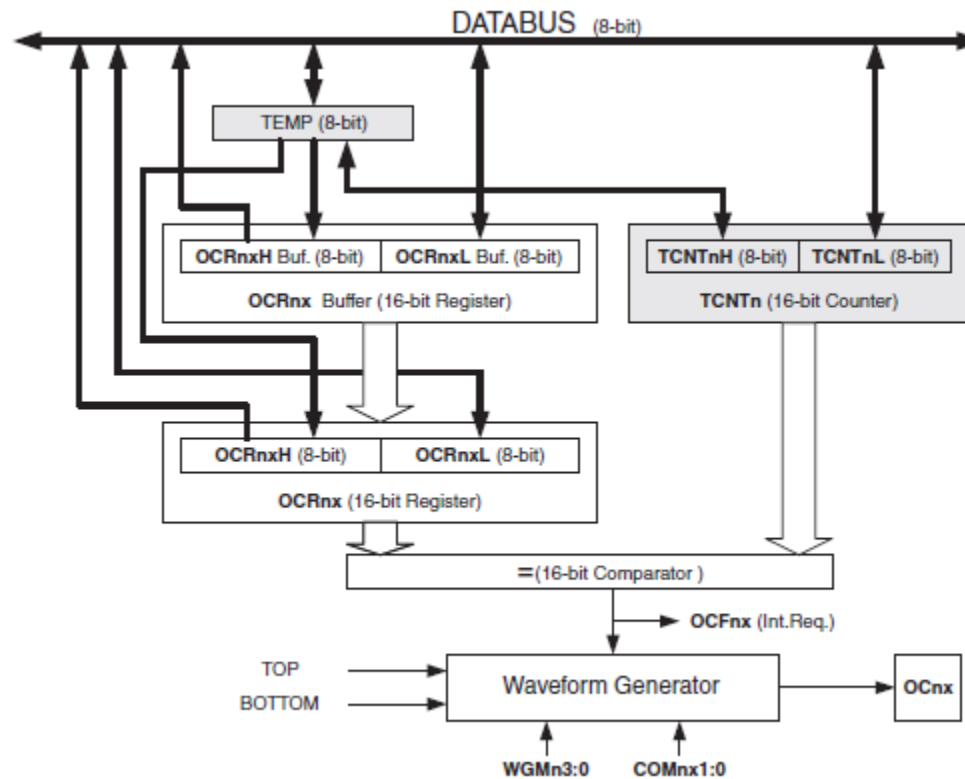
Pulse Width Modulation

◆ ATmega128 Timer/Count0 PWM



Pulse Width Modulation

◆ ATmega128 Timer/Count1 PWM



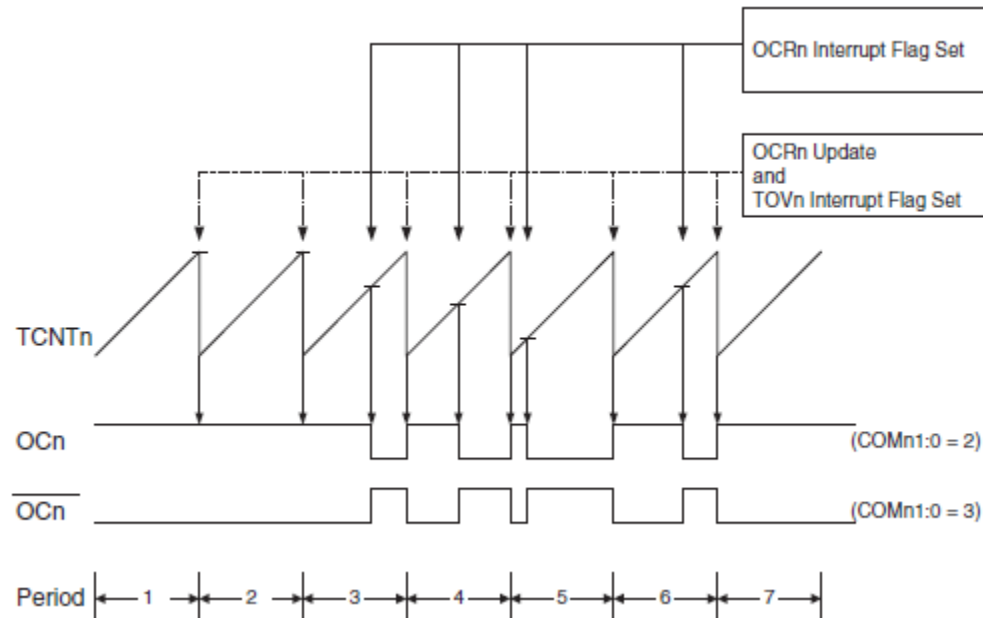
Pulse Width Modulation

◆ ATmega128

- Fast PWM Mode (고속 PWM)
 - 다른 PWM 모드에 비하여 약 2배의 주파수를 갖는다.
 - 단순한 PWM 제어에 사용
- Phase Correct PWM Mode (PC PWM)
 - Fast PWM 모드에 비하여 $\frac{1}{2}$ 로 낮은 주파수
 - 2배로 높은 분해능의 PWM 출력 신호를 발생
 - 고정된 주파수에서 아주 가끔 듀티를 가변하는 정도의 PWM 동작에 적합
- Phase and Frequency Correct PWM Mode (PFC PWM)
 - 16비트 타이머에서 사용 가능
 - 듀티비와 주파수를 모두 변경하여야 하는 응용에 적합

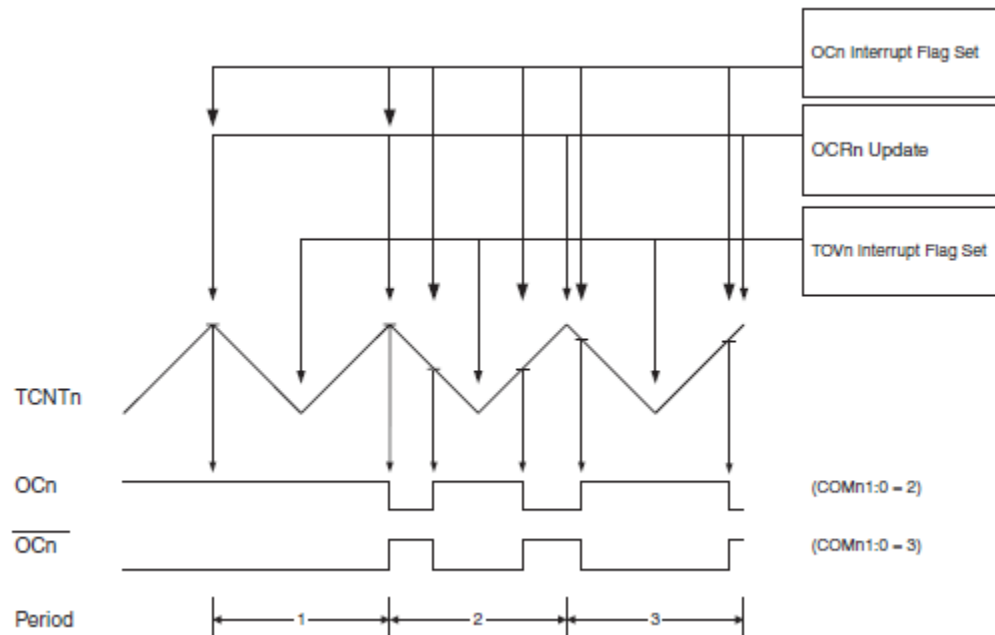
Pulse Width Modulation

◆ Fast PWM Mode



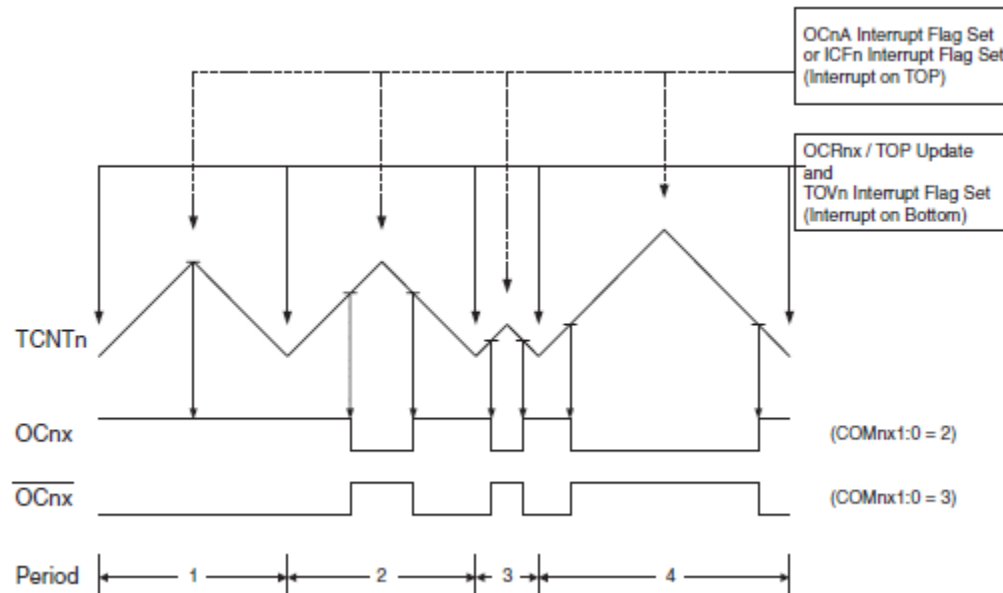
Pulse Width Modulation

◆ Phase Correct PWM Mode



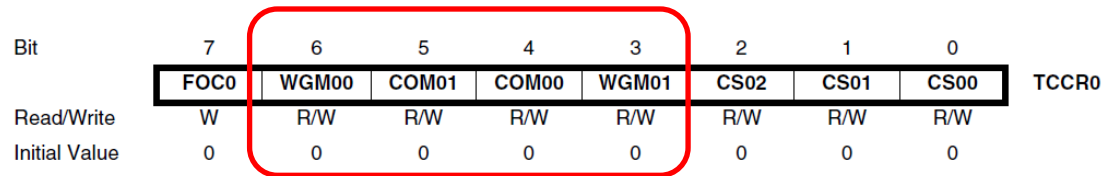
Pulse Width Modulation

◆ Phase and Frequency Correct PWM Mode



Pulse Width Modulation

◆ Register



Mode	WGM01 ⁽¹⁾ (CTC0)	WGM00 ⁽¹⁾ (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0 at	TOV0 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Fast PWM Mode

PC PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match when up-counting. Set OC0 on compare match when downcounting.
1	1	Set OC0 on compare match when up-counting. Clear OC0 on compare match when downcounting.

Pulse Width Modulation

Register

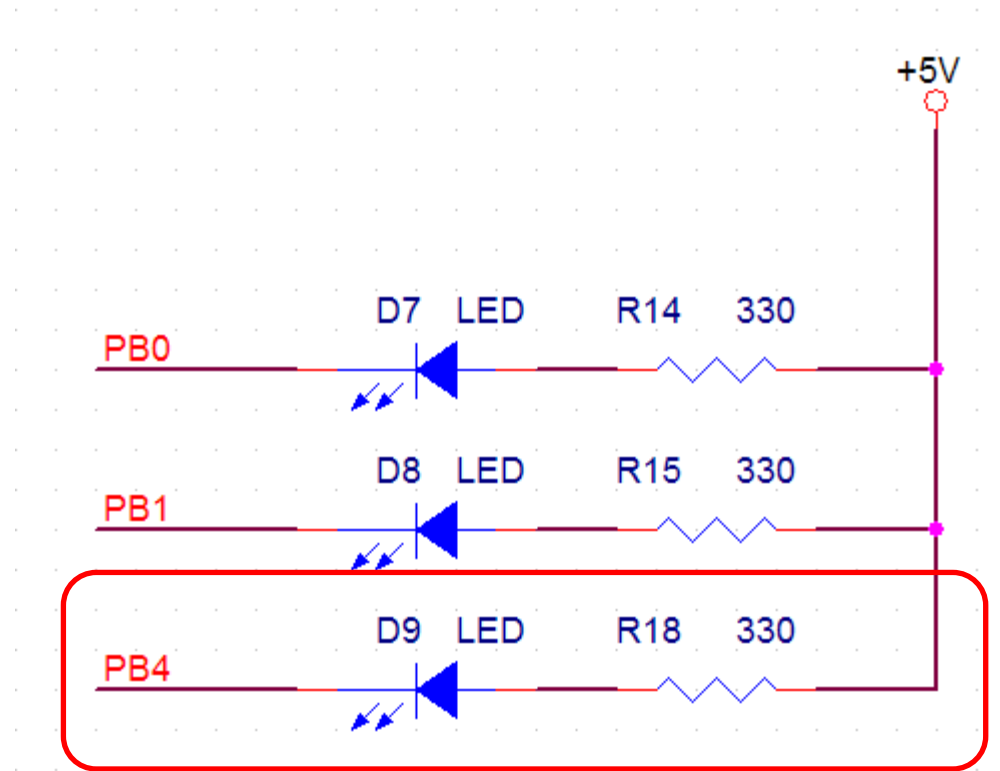
Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	—	—	—
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Pulse Width Modulation

◆ 추가 회로



OC0 Port에 해당하는 PORTB4에 LED를 연결한다.

Pulse Width Modulation

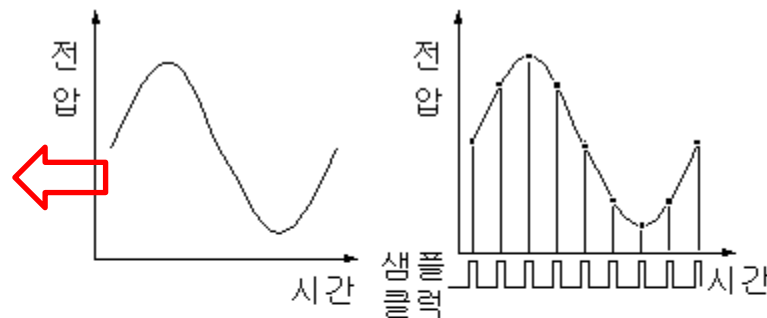
- `#include <mega128.h>`
- `#include <delay.h>`
- `unsigned char duty = 0;`
- `void main(void)`
- `{`
- `PORTB = 0x00;`
- `DDRB = 0x10;`
- `// PWM 출력 포트는 반드시 출력으로 설정되어야 한다.`
- `TCCR0 = 0x6F;`
- `// Prescaler 1024`
- `// Mode : Fast PWM`
- `// OC0 output`
- `while(TRUE)`
- `{`
- `duty++;`
- `OCR0 = duty;`
- `delay_ms(10);`
- `};`
- `}`

PORTB 4번 포트를 오실로스코프로 측정하면
주파수는 약61Hz로 듀티비가 0~100%로 변하게 된다.

Analog to Digital Converter

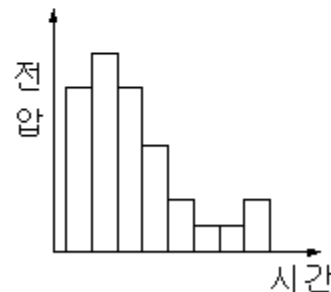
◆ ADC란?

연속적인 아날로그 신호
각종 센서들로부터 입력된 신호
ex) 전압, 전류, 온도, 소리

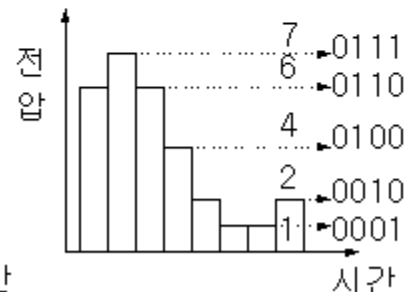


(a) Analog Signal

(b) Sampling



(c) Quantization



(d) Coding

유효한 값의 디지털 신호
ex) 0 or 1

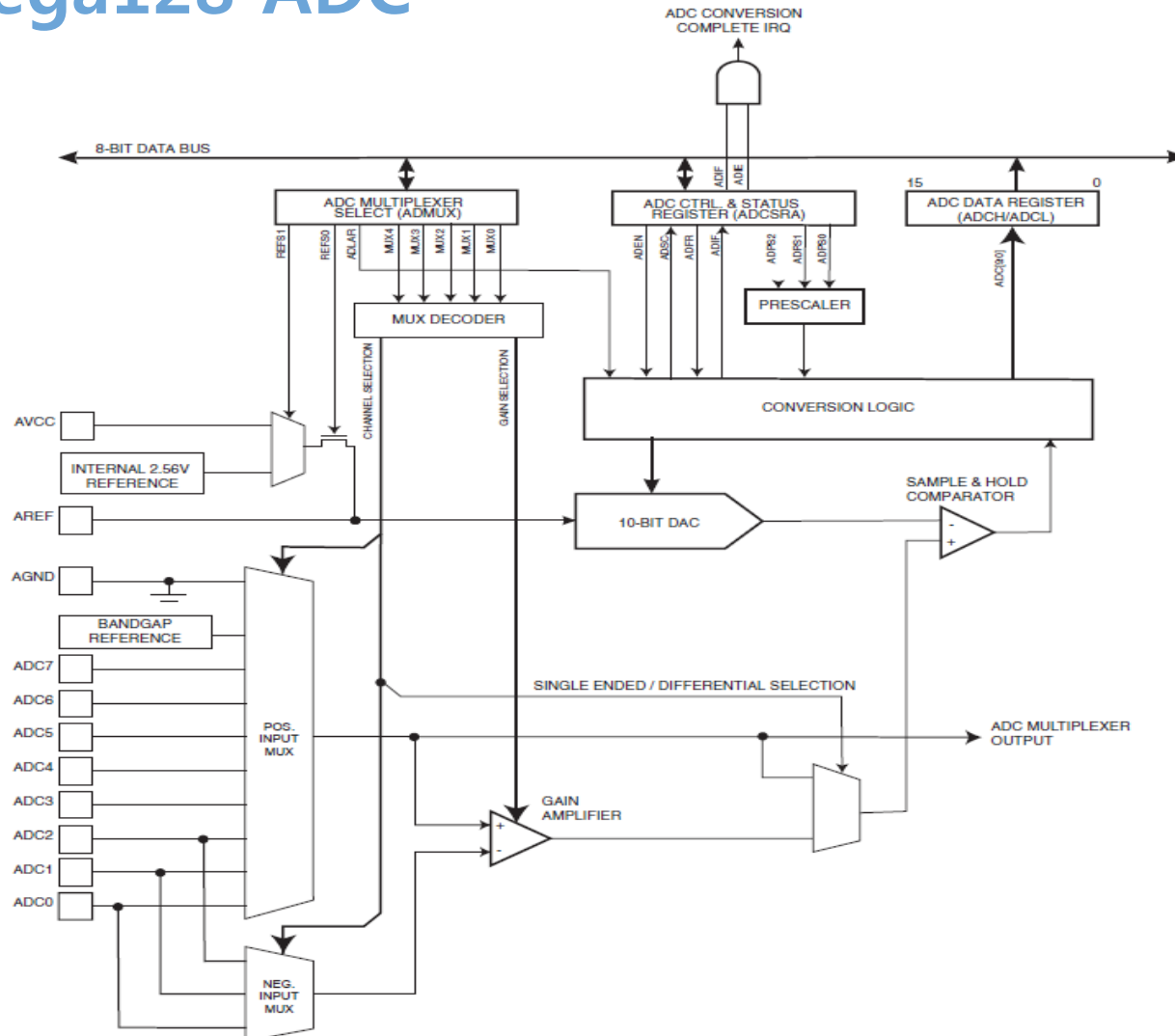
Analog to Digital Converter

◆ ATmega128 ADC

- 10-bit 분해능
- 12~260us의 변환 시간
- 초당 76,900 샘플링까지 가능(최대 분해능에서는 15ksps 가능)
- 8개의 멀티플렉스된 단극성 입력 채널
- 7개의 차동 입력 채널
- 10배와 200배의 선택적 이득을 갖는 두 개의 차동 입력 채널
- 0~VCC까지의 ADC 입력 전압 범위
- 프리 런닝 또는 단일 변환 모드
- ADC 변환 완료 시에 인터럽트 발생

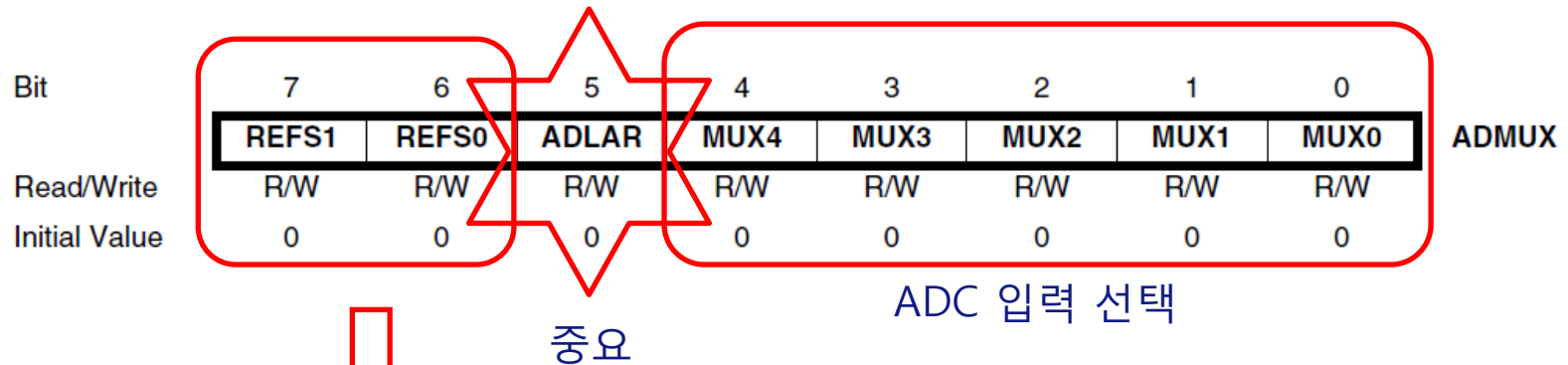
Analog to Digital Converter

◆ ATmega128 ADC



Analog to Digital Converter

◆ Register



기준 전압 선택

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Analog to Digital Converter

◆ Register

ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The ADC Data Register – ADCL and ADCH

ADLAR = 0:

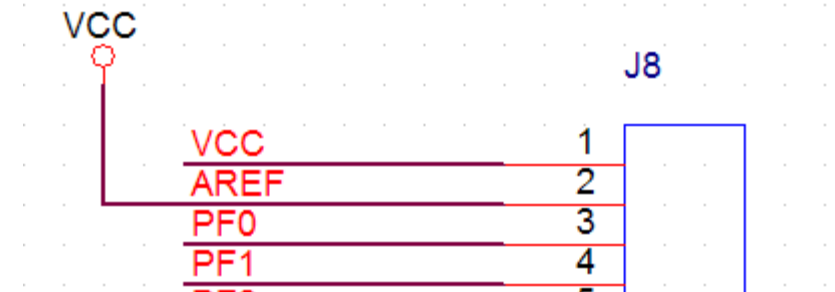
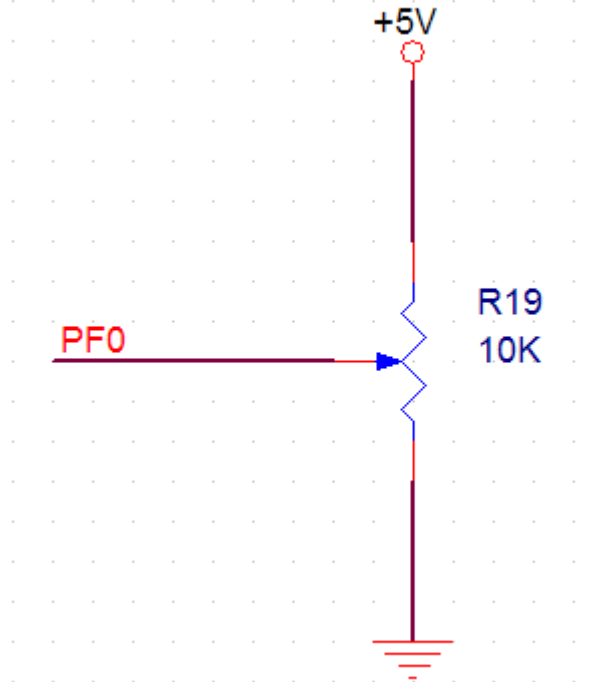
Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	

ADLAR = 1: 8bit로 사용이 가능

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

Pulse Width Modulation

◆ 추가 회로



ADC0번에 해당하는 PORTF 0번에 다음과 같이 가변저항을 연결한다.
VREF핀을 VCC(5V)에 연결한다.

Pulse Width Modulation

```
• #define ADC_VREF_TYPE 0x40
• unsigned int ADC = 0;

• unsigned int read_adc(unsigned char adc_input)
• {
•     ADMUX = adc_input | (ADC_VREF_TYPE & 0xff);
•     delay_us(10);

•     ADCSRA |= 0x40;

•     while ( ADCSRA & 0x10) == 0 );
•     ADCSRA |= 0x10;

•     return ADCW;
• }

• void main(void)
• {
•     DDRF = 0x00;
•
•     // ADC initialization
•     // ADC Clock frequency: 1000.000 kHz
•     // ADC Voltage Reference: AREF pin
•     ADMUX = ADC_VREF_TYPE & 0xff;
•     ADCSRA = 0x84;

•     while(TRUE)
•     {
•         ADC = read_adc(0);
•         // LCD에 변수 ADC출력
•     };
• }
```

Exercise

- ◆ PWM의 다른 2가지의 모드를 동작시켜 보고 차이점에 대해 생각해본다.
- ◆ ADC 완료되었을 때, 인터럽트가 발생하도록 하여 동작시켜 보시오.
- ◆ 가변 저항을 이용하여, LED의 밝기를 조절 하시오.
 - 가변저항을 돌렸을 때, 전압은 0~5V로 변하게 된다. 이를 ADC로 입력을 받는다.
 - 이에 따라 PWM을 발생 시켜 LED를 동작 시키면 밝기를 조절할 수 있다.

Thank You !

www.gajest.com